AN N JOB, SINGLE MACHINE SEQUENCING ALGORITHM FOR DECREASING THE SUM OF THE COMPLETION TIMES SUBJECT TO A MINIMUM NUMBER OF LATE JOBS

Richard Lee Gates

United States
Naval Postgraduate School

THESIS

An n Job, Single Machine Sequencing Algorithm
for Decreasing the Sum of the Completion Times
Subject to a Minimum Number of Late Jobs

by

Richard Lee Gates

Thesis Advisor: G. T. Howard

September 1971

An n Job, Single Machine Sequencing Algorithm
for Decreasing the Sum of the Completion Times
Subject to a Minimum Number of Late Jobs

by

Richard Lee Gates
Commander, United States Navy
B.S., Naval Postgraduate School, 1963

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1971

ABSTRACT

An n job, single machine sequencing algorithm is developed which decreases the sum of the completion times subject to a minimum number of late jobs. A primal approach is employed in which successively better solutions are obtained while maintaining feasibility. Optimality, while not claimed, may be achieved in some problems. Possible industrial and military applications are discussed.

TABLE OF CONTENTS

I. INTRODUCTION

A. NATURE OF THE PROBLEM

The problem considered in this thesis is one belonging to a general class of problems in which a finite set of jobs must be sequenced through a single facility, minimizing some function of lateness. Specifically, the problem is to sequence n jobs, $J_1,\ldots,J_n$, with known processing times, $p_1,\ldots,p_n$, and due-dates, $d_1,\ldots,d_n$, through a single production facility in such a way as to decrease the sum of the completion times, $C = \sum_{i=1}^{n} c_i$, subject to a minimum number of late jobs, U*. This criterion provides for more efficient scheduling by decreasing the average repair or service time of a unit while maintaining a minimum number of late jobs. This problem has not been treated in the literature to date.

The processing times, which are defined to include set-up and tear-down times, are independent of the sequence. The problem is static in that job arrivals subsequent to the n^{th} job are not considered and preemption is not permitted. It is assumed the jobs are processed continuously until all are completed with no lot-splitting permitted.

Problem formulation originated from research into message sequencing procedures utilized by shipboard communications centers. While this paper does not attempt to solve the general dynamic problem with its many priority restraints, the algorithm developed herein is applicable to

messages of a single priority class processed under static conditions.

The algorithm has given the optimal solution in all examples considered for $n \leq 10$ and is computationally feasible for small n yielding a solution manually in a few minutes. For large n it could easily be programmed for computer solution.

B. NOTATION AND DEFINITIONS OF TERMS

The following notation and definitions are employed:

S_v	- The v^{th} sequence of jobs	$S_v = (J_1,\ldots,J_n)$ $S=(A,R)$
c_i	- Completion time of job i	$c_i = p_i + W_i = c_{i-1} + p_i$
C_v	- Sum of completion times for sequence S_v	$C_v = \sum_{k=1}^{n} c_k$ $v=1,\ldots,9$
p_i	- Processing time of job i	
d_i	- Due-date of job i	
L_i	- Lateness of job i	$L_i = c_i - d_i$
T_i	- Tardiness of job i	$T_i = \max(0, L_i)$
E_i	- Earliness of job i	$E_i = \max(0, -L_i)$
W_i	- Waiting time of job i before processing begins	$W_i = \sum_{k=1}^{i-1} p_k$
s_i	- Slack time of job i	$s_i = d_i - p_i$
α	- Smallest numerical difference between processing times of any two jobs	$\alpha = \min_{\substack{i,i' \\ i \neq i'}} (p_{i'} - p_i)$
A	- Sequence of jobs forming a subset of S, whose completion times are all early	$A = (J_1,\ldots,J_{i-1},J_i,J_{i+1},\ldots,J_m)$
R	- Sequence of jobs forming a subset of S, whose completion times are all late	$R = (J_{m+1},\ldots,J_{j-1},J_j,J_{j+1},\ldots,J_n)$

J_i - An arbitrary early job i $\quad c_i \leq d_i$

J_j - An arbitrary late job j $\quad c_j > d_j$

U_v - Number of late jobs in sequence S_v

II. BACKGROUND

A. PREVIOUS KNOWN RESULTS

Research previously reported in this area is contained in [1] - [8]. When both processing times and due-dates are deterministic and known, the following results related to this problem have been proved:

1. The Shortest Processing Time Rule (SPT)

The average completion time $C = (1/n)\Sigma c_i$, and the average lateness $\overline{L} = (1/n)\Sigma L_i$, are minimized by scheduling jobs in order of increasing processing time. [7], [1, Theorem 3-2].

2. The Due Date Rule (DDATE)

Maximum job lateness and maximum job tardiness are minimized by scheduling the jobs in order of increasing due dates [3], [1, Theorem 3-3].

3. The Minimize Sum of Completion Times Subject to Number Late Jobs Algorithm

This algorithm will be referred to as Smith's algorithm [7], [1, Theorem 3-5] in this paper. If there exists a sequence such that maximum job tardiness is zero, then there is an ordering of the jobs with job K in the last position which minimizes mean completion time (subject to condition that maximum tardiness remain zero), if and only if:

a. $$d_k \geq \sum_{i=1}^{n} p_i$$

b. $$p_k \geq p_i \quad \forall i \text{ with } d_i \geq \sum_{k=1}^{n} p_k$$

4. <u>The Minimum Number of Late Jobs Algorithm</u>

This algorithm will be referred to as Hodgson's algorithm in this paper. J. M. Moore [6] has developed an algorithm for sequencing n jobs through a single facility to minimize the number of late jobs. A similar computationally faster algorithm has been developed by T. J. Hodgson [6], and proved in [8].

B. SMITH'S ALGORITHM

Smith's algorithm is a procedure which sequences in SPT order subject to the condition of no late jobs. It will be employed frequently in the algorithm of this paper. The schedule is developed recursively, by selecting a job for n, then n-1, etc. The job with the greatest processing time from the set of jobs with due dates not less than the completion time of the n^{th} job is selected for the n^{th} position. Once this position is filled, the completion time of job n-1 is known:

$$c_{n-1} = \sum_{k=1}^{n-1} p_k .$$

From the remaining jobs, the job with the largest processing time from the set of jobs with due dates not less than c_{n-1} is selected for position n-1. Consider the following example, ordered according to DDATE Rule to insure no late jobs.

	J_1	J_2	J_3	J_4	J_5	
p_i	2	5	3	5	4	$\Sigma p_i = 19$
d_i	4	10	12	20	23	
c_i	2	7	10	15	19	$\Sigma c_i = 53$

The sum of the processing times is 19. Only jobs J_4 and J_5 have due dates not less than 19. J_4 is chosen for position 5 since $p_4 > p_5$. Subtracting $p_4=5$ from 19 results in $c_{n-1}=14$. Excluding J_4, only J_5 has a due date not less than 14 and therefore it is chosen for position 4. Continuing in similar manner J_2 is chosen for position 3 as $p_2 > p_3 > 9=c_{n-2}$, J_3 for position 2 and J_1 for position 1. The final sequence is:

	J_1	J_3	J_2	J_5	J_4	
p_i	2	3	5	4	5	
d_i	4	12	10	23	20	
c_i	2	5	10	14	19	$\Sigma c_i=50.$

C. HODGSON'S ALGORITHM

The version of this algorithm by Hodgson is employed in this paper to find the minimum number of late jobs and to provide an initial feasible solution to start the algorithm improvement procedure. Steps in Hodgson's algorithm [6] are:

Step One. Order the jobs according to their due-dates, $d_1 \leq \cdots \leq d_n$, and call the resulting ordering the current sequence, $J_1, \ldots, J_n$.

Step Two. Using the current sequence, find the first late job J_q, and go to Step Three. If no such job is found, the algorithm terminates with an optimal schedule obtained by placing those jobs that have been rejected after the jobs in the current sequence in any order.

Step Three: Find the job in the subsequence, $J_1,\ldots,J_q$, of the current sequence having the largest processing time and reject it from the current sequence. Return to Step Two, using the resulting sequence as the current sequence.

An example of this procedure follows:

Step One. Order by due-dates.

	J_1	J_2	J_3	J_4	J_5	J_6
p_i	2	1	5	7	6	3
d_i	2	4	5	14	15	28
c_i	2	3	8	15	21	24

Step Two. Find first late job. It is J_3 ($c_3 > d_3$). Go to Step Three.

Step Three. The job with the greatest processing time in the subsequence (J_1,J_2,J_3) is J_3 with $p_3=5$. Therefore, reject it and return to Step Two with the new current sequence.

Step Two. The schedule is now:

	Current Sequence					Rejected Jobs
	J_1	J_2	J_4	J_5	J_6	J_3
p_i	2	1	7	6	3	5
d_i	2	4	14	15	28	5
c_i	2	3	10	16	19	24

The first late job is J_5. Go to Step Three.

Step Three. The job with the greatest processing time J_q in subsequence ($J_1,\ldots,J_5$) is J_4 with $p_4=7$. Therefore, reject it and return to Step Two with new current sequence.

Step Two. The schedule is now:

	Current Sequence				Rejected Jobs	
	J_1	J_2	J_5	J_6	J_3	J_4
p_i	2	1	6	3	5	7
d_i	2	4	15	28	5	14
c_i	2	3	9	12	17	24

The algorithm now terminates as there are no remaining late jobs in the current sequence. The above schedule is now optimal with the current sequence followed by the rejected jobs in any order. The minimum number of late jobs is $U^*=2$.

III. ALGORITHM

A. DESCRIPTION

The problem can be restated as follows: Given a set of jobs numbered 1 to n find the sequence $S^* = (J_1,\ldots,J_n)$ to

$$\min \sum_{k=1}^{n} c_k$$

Subject to: $U_v = U^*$ Where U^* is the minimum number of late jobs.

The algorithm employs a primal approach maintaining a feasible solution and it successively decreases the sum of the completion times C with successive sequences $S_v, v=1,\ldots,9$, by application of one of the algorithm's rules and/or Smith's algorithm, terminating with the sequence S_b when no further improvement can be accomplished. Hodgson's algorithm is employed to determine U^*, and to provide the initial feasible sequence. Rules are employed to interchange or insert jobs decreasing C at each step. Whenever the sequence $S = (A,R)$ results, C is improved by sequencing A by Smith's algorithm and R by the SPT rule. The algorithm terminates when application of the rules yield no further reduction in C. Possible optimality is checked at each step for early algorithm termination.

There exist at least three possible ways to decrease C from Hodgson's solution. The first, called the Interchange Rule, exchanges a late job $J_j \varepsilon R$ with an early job $J_i \varepsilon A$ where $p_j < p_i$ and $J_j \varepsilon A$ is now early. The second,

called the Insertion Rule, inserts a late job $J_j \varepsilon R$ into the A subsequence, $J_j \varepsilon A$ remaining late and all other jobs $J_i \varepsilon A$ remaining early. For application of this insertion there must exist at least one $J_i \varepsilon A$ with $p_i > p_j$ and a positive reduction in C must result. The third, called the SPT-Interchange Rule, examines the possibility of improvement when there exists a $J_j \varepsilon R$ whose $p_j > p_i \forall J_i \varepsilon A$. The need for such a rule is illustrated by example three, page 22. This last rule covers job interchanges not eligible under the Interchange Rule and necessitates some reordering of A if any reduction in C is possible. Subsequence A is reordered in SPT creating subsequence A' with one or more late jobs $J_i' \varepsilon A'$. Job $J_i' \varepsilon A'$ is then interchanged with job $J_j \varepsilon R$ if job $J_j \varepsilon A'$ is now early. Specific requirements of the rules are covered in paragraph B.

The rules are applied in the order given above to utilize the advantages of the partitioned sequence S = (A,R). Possible candidates for interchange are readily determined and Smith's algorithm and SPT can be applied to A and R respectively to improve C. For further improvement a branching occurs, with either the Insertion or SPT-Interchange the applicable rule. In the former, the partitioning scheme is destroyed while the latter maintains S = (A,R) and covers those interchanges not possible under the Interchange Rule. By reducing C at each step the number of possible sequences with $C \leq C_v$ is reduced, with each successive S_v approaching more closely to the optimal solution, S*, and C*, U*.

B. RULES

The following rules are employed in the algorithm:

1. Interchange Rule

a. Find the smallest processing time p_j among jobs in R.

b. Find the first job in A such that its processing time p_i satisfies $p_j < p_i$ and $c_{i-1} \leq s_j$.

c. Interchange J_i and J_j.

2. Insertion Rule

Provided there exists at least one $J_i \varepsilon A \ni p_i > p_j$, insert job J_j from R, into A between J_{i-1} and J_i where i maximizes $\sum_{k=i}^{j-1} (p_k - p_j) > 0$, the reduction in C.

Subject to:

$$
\begin{array}{lll}
c_{i-1} & > s_j & (J_j \text{ remains late in A)} \\
c_{i-1} + p_j & \leq s_i & \text{(All other jobs in A remain early)} \\
c_{i-1} + p_j + p_i & \leq s_{i+1} & \\
\vdots & \vdots & \\
c_{i-1} + p_j + \sum_{k=i}^{m-1} p_k & \leq s_m &
\end{array}
$$

Continue until no further insertions are possible.

3. SPT-Interchange Rule

If there exists a $J_j \varepsilon R$ such that $p_j > p_i$ for all jobs in A, then:

a. Reorder A in SPT. Call the resulting subsequence A'.

b. Determine number of late jobs in A' = $U_{A'}$.

c. If $U_{A'} \leq U^*$, interchange $J_j \varepsilon R$ and $J_i^! \varepsilon A'$ if:

$$c'_{i-1} \leq s_j \text{ and } C \text{ is decreased.}$$

Continue interchanges if possible until $U_{A'} = 0$.

C. STEPS

Steps in the algorithm are as follows:

1. Step One

Order the jobs in accordance with the SPT rule. $(p_1, \leq \ldots \leq, p_n)$. Determine the number of late jobs, U_1, and $\Sigma c_i = C_1$. If there are no late jobs the sequence $S_1 = (J_1, \ldots, J_n)$ is optimal and the algorithm terminates with $C^* = C_1, U^* = 0$.

2. Step Two

Reorder the jobs in accordance with Hodgson's algorithm. Call the resulting sequence $S_2 = (A,R)$ where all jobs in A are early and all jobs in R are late. Determine C_2 and $U_2 = U^*$. If $U^* = U_1$, the algorithm terminates and the sequence S_1 is optimal with $C^* = C_1$ and $U^* = U_1$. The current sequence is then:

$$S = (J_1, \ldots, J_{i-1}, J_i, J_{i+1}, \ldots, J_m | J_{m+1}, \ldots, J_{j-1}, J_j, J_{j+1}, \ldots, J_n)$$

3. Step Three

Reorder A and R by Smith's algorithm and SPT respectively. Call the resulting sequence $S_3 = (A_3, R_3)$. Determine C_3. If $C_3 = C_1 + \alpha$, S_3 is optimal and the algorithm terminates with $C^* = C_3$.

4. Step Four

Interchange jobs in accordance with the interchange rule. If unable to do this go to Step Six. Call the resulting

sequence $S_4 = (A_4,R_4)$. Determine C_4. $U_4 = U^*$. If $C_4 = C_1+\alpha$, S_4 is optimal and the algorithm terminates with $C^* = C_4$.

5. Step Five

Reorder A_4 and R_4 by Smith's algorithm and SPT respectively. Call the resulting sequence $S_5 = (A_5,R_5)$. Determine C_5. $U_5 = U^*$. If $C_5 = C_1+\alpha$, S_5 is optimal and the algorithm terminates with $C^* = C_5$.

6. Step Six

Insert jobs in accordance with the insertion rule. If unable to do this go to Step Seven. Call the resulting sequence S_6. Determine C_6. $U_6 = U^*$. If $C_6 = C_1+\alpha$, S_6 is optimal and the algorithm terminates with $C^* = C_6$. Go to Step Nine.

7. Step Seven

Conduct interchanges in accordance with the SPT-interchange rule. If unable to do this go to Step Nine. Call the resulting sequence $S_7 = (A_7,R_7)$. Determine C_7. $U_7 = U^*$. If $C_7 = C_1+\alpha$, S_7 is optimal and the algorithm terminates with $C^* = C_7$.

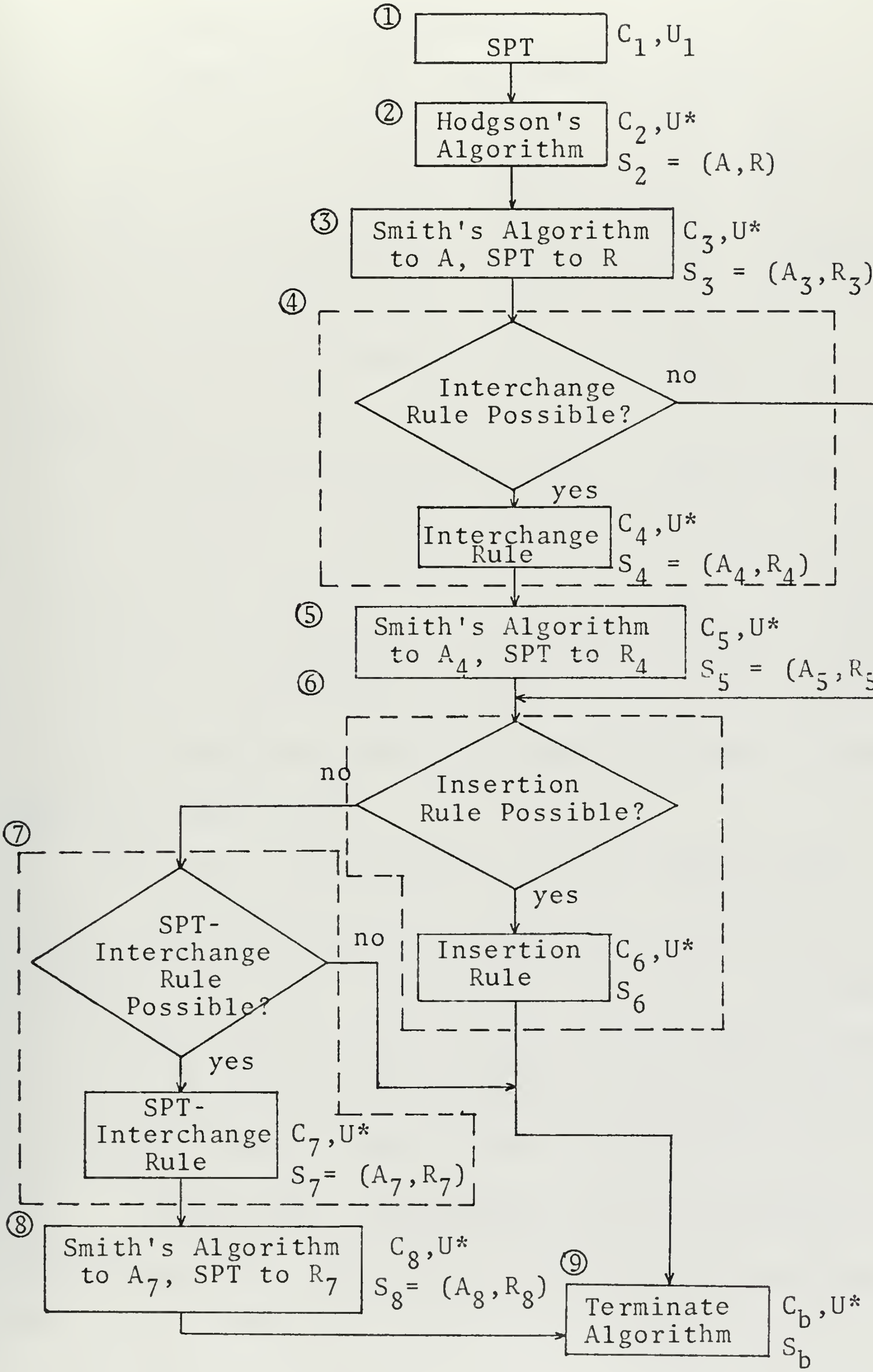
8. Step Eight

Reorder A_7 and R_7 by Smith's algorithm and SPT respectively. Call the resulting sequence $S_8 = (A_8,R_8)$. Determine C_8. $U_8 = U^*$. If $C_8 = C_1+\alpha$, S_8 is optimal and the algorithm terminates with $C^* = C_8$.

9. Step Nine

The algorithm terminates with S_3, S_5, S_6 or S_8, being S_b. C_b is an upper feasible bound on C* determined by $C_b = \min(C_3, C_5, C_6, C_8)$. $C_b \geq C^*$.

D. BLOCK DIAGRAM OF ALGORITHM



E. EXAMPLES

1. An Interchange Solution ($C_b = C_5 = C^*$)

Step One. Order the sequence in SPT.

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	
p_i	4	5	6	7	8	9	9	10	10	12	
d_i	55	50	45	60	58	39	67	54	67	30	$U_1=3(J_8,J_9,J_{10})$
c_i	4	9	15	22	30	39	48	58	68	80	$C_1=373$

Step Two. Reorder jobs in accordance with Hodgson's algorithm.

	A								R		
	J_{10}	J_6	J_3	J_2	J_8	J_1	J_5	J_7	J_4	J_9	
p_i	12	9	6	5	10	4	8	9	7	10	$S_2=(A,R)$
d_i	30	39	45	50	54	55	58	67	60	67	$U^*=2(J_4,J_9)$
c_i	12	21	27	32	42	46	54	63	70	80	$C_2=447$

Step Three. Reorder A and R by Smith's algorithm and SPT respectively.

	A_3								R_3		
	J_1	J_2	J_{10}	J_6	J_3	J_8	J_5	J_7	J_4	J_9	
p_i	4	5	12	9	6	10	8	9	7	10	$S_3=(A_3,R_3)$
d_i	55	50	30	39	45	54	58	67	60	67	$U^*=2(J_4,J_9)$
c_i	4	9	21	30	36	46	54	63	70	80	$C_3=413$
s_i	51	45	18	30	39	44	50	58	53	57	

Step Four. Interchange jobs in accordance with the interchange rule. The first candidate for interchange is J_4 with $p_j = 7$. The first $J_i \varepsilon A$ satisfying $p_j < p_i$ and $c_{i-1} \leq s_j$ is J_{10} with $p_j = 7 < p_i = 12$ and $c_{i-1} \leq s_j$ is J_{10} with

$p_j = 7 < p_i = 12$ and $c_{i-1} = 9 \leq s_j = 53$. The results of interchanging J_4 and J_{10} are:

	$\leftarrow A_4$							$\rightarrow$	$\leftarrow R_4$	$\rightarrow$	
	J_1	J_2	J_4	J_6	J_3	J_8	J_5	J_7	J_{10}	J_9	
p_i	4	5	7	9	6	10	8	9	12	10	
d_i	55	50	60	39	45	54	58	67	30	67	$U^*=2(J_{10},J_9)$
c_i	4	9	16	25	31	41	49	58	70	80	$C'=383$
s_i	51	45	53	30	39	44	50	58	18	57	

There remains no $J_j \varepsilon R$ with a $p_j < p_i \forall J_i \varepsilon A_4$ and the interchanges end with $C_4 = C' = 383$, $S_4 = (A_4, R_4)$.

Step Five. Reorder A_4 and R_4 by Smith's algorithm and SPT respectively. The result is:

	$\leftarrow A_5$							$\rightarrow$	$\leftarrow R_5$	$\rightarrow$	
	J_1	J_2	J_3	J_4	J_5	J_6	J_8	J_7	J_9	J_{10}	
p_i	4	5	6	7	8	9	10	9	10	12	$S_5=(A_5,R_5)$
d_i	55	50	45	60	58	39	54	67	67	30	$U^*=2(J_9,J_{10})$
c_i	4	9	15	22	30	39	49	58	68	80	$C_5=374=C_1+\alpha=C^*$

Since $C_5 = C^*$ the algorithm terminates with S_5 the optimal sequence.

2. <u>An Insertion Solution</u> ($C_b = C_6 = C^*$)

Step One. Order the sequence in accordance with SPT rule.

	J_1	J_2	J_3	J_4	J_5	
p_i	1	2	3	4	5	
d_i	6	1	14	7	20	$U_1=2(J_2,J_4)$
c_i	1	3	6	10	15	$C_1=35$

Step Two. Reorder the jobs in accordance with Hodgson's algorithm.

	A				R	
	J_1	J_4	J_3	J_5	J_2	
p_i	1	4	3	5	2	$S_2=(A,R)$
d_i	6	7	14	20	1	$U^*=1(J_2)$
c_i	1	5	8	13	15	$C_2=42$
s_i	5	3	11	15	-1	

Step Three. Reorder A and R by Smith's algorithm and SPT respectively. The result is identical with $S_2 = S_3 = (A_3,R_3)$.

Step Four. Interchange jobs in accordance with the interchange rule. Since S_2 is negative and all $c_{i-1}>0$, J_2 does not qualify for interchange. Go to Step Six.

Step Six. Insert jobs in accordance with insertion rule. The insertion location is determined in part by finding the i which maximizes $[\sum_{k=i}^{j-1}(p_k-p_j)]>0$, the reduction in C. These quantities are:

Insert between	$\sum_{k=i}^{j-1}(p_k-p_j)$	
$0, J_1$	$(-1+2+1+3) = 5$	
J_1, J_4	$(2+1+3) = \underline{6}$ (max) $\Rightarrow$	insert J_2 between J_1 & J_4 if lateness conditions are met
J_4, J_3	$(1+3) = 4$	
J_3, J_5	$(3) = 3$	

Check if J_2, the only member of R_3, will still be late if inserted and if all jobs $J_i,\ldots,J_m$ will remain early.

Job	Condition to Satisfy	Result	
$J_j = J_2$	$c_{i-1} > s_j$	$1>-1$	$\Rightarrow J_2$ late
$J_i = J_4$	$c_{i-1}+p_j \leq s_i$	$1+2\leq 3$	$\Rightarrow J_4$ early
$J_{i+1}=J_3$	$c_{i-1}+p_j+p_i \leq s_{i+1}$	$1+2+4\leq 11$	$\Rightarrow J_3$ early
$J_m = J_5$	$c_{i-1}+p_j+\sum_{k=i}^{m-1} p_k \leq s_m$	$1+2+(4+3)\leq 15$	$\Rightarrow J_5$ early

Since J_2 inserted between J_1 and J_4 maximizes the reduction in C by 6 and also satisfies the constraints, there is no need to check any of the other insertion points. The result is:

	J_1	J_2	J_4	J_3	J_5	
p_i	1	2	4	3	5	$S_6=(A_6,R_6)$
d_i	6	1	7	14	20	$U^*=1(J_2)$
c_i	1	3	7	10	15	$C_6=36=C_1+\alpha=C^*$

Since $C_6 = C^*$ the algorithm terminates with S_6 the optimal sequence.

3. <u>A SPT-Interchange Solution</u> ($C_b = C_8 = C^*$)

Step One. Order the sequence in SPT.

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	
p_i	1	1	1	3	3	3	4	
d_i	1	5	6	4	10	14	9	$U_1=2(J_4,J_7)$
c_i	1	2	3	6	9	12	16	$C_1=49$

Step Two. Reorder jobs in accordance with Hodgson's algorithm.

	←		A			→	R	
	J_1	J_4	J_2	J_3	J_5	J_6	J_7	
p_i	1	3	1	1	3	3	4	
d_i	1	4	5	6	10	14	9	$U^*=1(J_2)$
c_i	1	4	5	6	9	12	16	$C_2=53$

Step Three. Reorder A and R by Smith's algorithm and SPT respectively. The result is identical with $S_2 = S_3 = (A_3,R_3)$ and $C_3 = C_2 = 53$.

Step Four. Interchange jobs in accordance with interchange rule. Since $p_j = 4 > p_i \forall J_i \varepsilon A$ no interchanges are permitted. Go to Step Six.

Step Six. Insert jobs in accordance with insertion rule. Since there is not a $p_i > p_j$ no insertions are permitted. Go to Step Seven.

Step Seven. Conduct interchanges in accordance with SPT-interchange rule. Since there exists a $J_j \varepsilon R$ such that $p_j > p_i \forall J_i \varepsilon A$, proceed as follows:

a. Reorder A in SPT. = A'.

	←		A'			→	R	
	J_1	J_2	J_3	J_4	J_5	J_6	J_7	
p_i	1	1	1	3	3	3	4	
d_i	1	5	6	4	10	14	9	$U_A'=1(J_4),U=2(J_4,J_7)$
c_i	1	2	3	6	9	12	16	
s_i	0	4	5	1	7	11	5	

b. Determine number of late jobs in $A' = U'_A = 1(J_4)$.

c. Since $U'_A = 1 \leq U^* = 2$, attempt to interchange $J_j = J_7$ and $J'_i = J_4$ if $c'_{i-1} \leq s_j$ and $C_7 < C_3$. $c'_{i-1} \leq s_j$ is satisfied as $3 \leq 5$. Checking the results of interchanging J_4 and J_7 to determine if $C_7 < C_3$ yields sequence:

	A_7						R_7	
	J_1	J_2	J_3	J_7	J_5	J_6	J_4	
p_i	1	1	1	4	3	3	3	$S_7=(A_7,R_7)$
d_i	1	5	6	9	10	14	4	$U^*=1(J_4)$
c_i	1	2	3	7	10	13	16	$C_7=52$, $C_7<C_3=53$

No further interchanges under this rule are required as $U'_A = 0$.

Step Eight. Reorder A_7 and R_7 by Smith's algorithm and SPT respectively. The result is identical with $S_7 = S_8 = (A_8,R_8)$ and $C_7 = C_8 = 52$.

Step Nine. The algorithm terminates with $C_b = \min(C_3,C_8) = \min(53,52)$. Therefore $S_8 = (A_8,R_8)$ is the sequence with the smallest $C = 52$. In this particular case $C_b = C^* = 52$, which may be proved by evaluating all the possible sequences.

F. THEORETICAL DEVELOPMENT

Beginning at Step Two, a primal approach is utilized in which successively better solutions are obtained while maintaining feasibility. It will be shown below how this is accomplished.

1. Consider a sequence $S_1 = (J_1, \ldots, J_n)$ ordered by increasing processing times, $(p_1, \leq \ldots \leq, p_n)$. By Smith [7], the SPT sequence is optimal with respect to minimizing $\Sigma c_i = C_1$. C_1 is the lower bound on C^*, the sum of the completion times for the optimal sequence. If the number of late jobs, U_1, equals zero then S_1 is optimal with $C_1 = C^*$.

2. Consider a sequence, S_2, ordered in accordance with Hodgson's algorithm:

$$S_2 = (\overbrace{J_1, \ldots, J_{i-1}, J_i, J_{i+1}, \ldots, J_m}^{A_2} \mid \overbrace{J_{m+1}, \ldots, J_{j-1}, J_j, J_{j+1}, \ldots J_n}^{R_2})$$

$$= (A_2, R_2)$$

where all jobs εA_2 are early $(c_i \leq d_i)$ and all jobs εR_2 are late $(c_j > d_j)$. Let $C_{A2} = \sum_{i=1}^{m} c_i, C_{R2} = \sum_{j=m+1}^{n} c_j$ and $C_2 = C_{A2} + C_{R2}$. By Hodgson [6], S_2 is optimal with respect to minimum number of late jobs, $U^* = U_2$. Therefore S_2 is a feasible solution which may be used as a starting sequence in the algorithm. If $U^* = U_1$, then the previous sequence S_1 is optimal as there is no other sequence with $C < C_1$ [1].

3. Since the maximum job tardiness of $A_2 = 0$, Smith's algorithm [7] can be used to minimize $\sum_{k=1}^{m} c_k$ in A_2 maintaining the number of late jobs in subsequence A_2 at 0. Call the resulting subsequence A_3 with min $\sum_{i \varepsilon A} c_i = C_{A3}$. R_2 is then reordered in SPT. Call the resulting subsequence R_3 and $\sum_{j \varepsilon R} c_j = C_{R3}$. Now $C_{A3} \leq C_{A2}$ and $C_{R3} \leq C_{R2} \Rightarrow C_{A3} + C_{R3} \leq C_{A2} + C_{R2} \Rightarrow C_3 \leq C_2$.

4. Step Four is the application of the interchange rule. It will be shown that resulting sequence S_4 improves S_3 with $C_4 \leq C_3$, maintaining $U = U^*$ constant. Conditions for the interchange of $J_j \varepsilon R$ and $J_i \varepsilon A$ are $p_j < p_i$ and $c_{i-1} \leq s_j$. Consider the contrary. Suppose job J_j with $p_j > p_i$ was interchanged with $J_i \varepsilon A$. This would increase C_3 by amount $(p_j - p_i)(j-i)$. If $p_j = p_i$, C_3 is unaffected. Therefore, C_3 is decreased only by interchanging J_j and J_i with $p_j < p_i$. Suppose an interchange resulted in $c_{i-1} > s_j = d_j - p_j$. But, then $J_j \varepsilon A$ is late. Since J_i now εR is late by [6] this implies both J_j and J_i are late. Therefore, the sequence can only be improved in C, with U constant, if $p_j < p_i$ and $c_{i-1} \leq s_j$. Continuing interchanging eligible candidates will decrease C by amount $(p_i - p_j)(j-i)$ on each interchange until no more interchanges are possible. Therefore, the resultant sequence $S_4 = (A_4, R_4)$ will result in $C_4 \leq C_3$.

5. Proof of Step Five is similar to the argument given in Step Three above. The resultant sequence $S_5 = (A_5, R_4)$ will yield $C_5 \leq C_4$.

6. Another possibility for improvement in C is by inserting a job $J_j \varepsilon R$ into A between J_{i-1} and J_i, job J_j remaining late while jobs $J_i, \ldots, J_m$ remain early. The insertion rule requirements are: existence of at least one $J_i \varepsilon A$ with $p_i > p_j$; where i maximizes $\sum_{k=i}^{j-1} (p_k - p_j) > 0$ (amount of reduction in C), and the following lateness conditions must be met:

$$
\begin{array}{lll}
c_{i-1} & > s_j & (J_j \varepsilon A \text{ remains late}) \\
c_{i-1}+p_j & \leq s_i & (J_i \text{ remains early}) \\
c_{i-1}+p_j+p_i & \leq s_{i+1} & (J_{i+1} \text{ remains early}) \\
\vdots & \vdots & \vdots \\
c_{i-1}+p_j+\sum_{k=i}^{m-1} p_k & \leq s_m & (J_m \text{ remains early})
\end{array}
$$

Now suppose the insertion of some job $J_j \varepsilon R$ into A was such that it did not satisfy one of the above conditions. If there were no job $J_i \varepsilon A \ni p_i > p_j$, this would imply $\sum_{k=i}^{j-1} (p_k - p_j) < 0 \,\forall\, i$, since $(p_i - p_j) < 0 \,\forall\, i$ results in an increase in C. Suppose some other insertion location is utilized, i.e., i is such that $\sum_{k=i}^{j-1} (p_k - p_j)$ is not a maximum and/or is negative in value. If the reduction is negative then there is no improvement in C. If the reduction is not a maximum, either it is not the best reduction possible, or the lateness conditions were not satisfied previously at other positions in A which had a higher value of $\sum_{k=i}^{j-1} (p_k - p_j)$ than position i. Therefore i must maximize the reduction in C subject to the lateness constraints.

If $c_{i-1} \leq s_j$, then $J_j \varepsilon A$ would have been early and qualified for interchange with J_i in Step Four. But, all such interchanges have taken place. Therefore, $J_j \varepsilon A$ must be late. If some arbitrary job $J_h \varepsilon (J_i, \ldots, J_m)$ is such that $c_{i-1}+p_j+\sum_{k=i}^{h-1} p_k > s_h \Rightarrow J_h$ is late, increasing U by one. Therefore, no job $J_j \varepsilon R$ exists for insertion except it satisfies all of the above conditions. These insertions continue until no further candidates remain.

Step Six was reached from Step Five or Three (if no interchanges were possible). Since each insertion has reduced C by $\sum_{k=i}^{j-1} (p_k - p_j) \Rightarrow C_6 < C_5$ or $C_6 < C_3$. At this stage of the solution the partitioned sequence S = (A,R) no longer exists and possible further improvement by interchanges between A and R are not meaningful. Therefore, upon completion of the sequence S_6 no further improvement is attempted and the algorithm goes to Step Nine.

7. As was explained on page 13, a third possibility for improvement in C is required beyond that of singular interchanges. One such method is called the SPT-Interchange rule. This rule reorders A in SPT to reduce C while creating additional late jobs in A which become candidates for singular interchanges with jobs $J_j \varepsilon R$. The reordered A subsequence is called $A' = (J_1, \ldots, J_i', \ldots, J_u, \ldots, J_m)$, where J_i' is a late job.

The first required condition is existence of a $J_j \varepsilon R \ni p_j > p_i \forall J_i \varepsilon A$. Suppose there is a $J_j \varepsilon R \ni p_j \leq p_i \forall J_i \varepsilon A$. If $J_j \varepsilon R \ni p_j < p_i$, then J_j was a candidate for interchange under the interchange rule of Step Four. But, all $J_j \varepsilon R$ have already been interchanged unless they would have been late in A. Therefore, any remaining $J_j \varepsilon R$ which could be considered for improvement must have $p_j \geq p_i \forall J_i \varepsilon A$. If $p_j = p_i$ there is no reduction in C by interchange. Therefore, J_j is such that $p_j > p_i \forall J_i \varepsilon A$.

The SPT-Interchange rule then requires $U_A' \leq U^*$. Suppose $U_A' \geq U^*$. This implies there are insufficient jobs εR

to interchange with $J_i' \varepsilon A'$ as U would be increased by amount $U_A'=U^*$, destroying feasibility. Interchanges are then conducted only if $c_{i-1}' \leq s_j$ and $C_7 < C_5$ (or C_3 if steps four and five were omitted). If $c_{i-1}' > s_j$ then both J_j and J_i' would be late. Finally, if $C_7 \geq C_5$ (or C_3) no improvement in C has taken place by this procedure. Continue interchanges until $U_A' = 0$. If unable to do this, interchange under this rule it is not permitted. If a positive reduction in C_5 (or C_3) results from this procedure the interchange is made. The resultant sequence is called S_7 with $C_7 < C_5$ (or C_3).

8. Proof of Step Eight is similar to the argument given in Step Three above. The resultant sequence $S_8 = (A_8, R_8)$ will yield $C_8 \leq C_7$.

9. Step Nine serves to terminate the algorithm if it has not terminated earlier due to C* being obtained. Any sequence with U* late jobs yielding a $C = C_1+\alpha$ is optimal as α is the smallest increment whereby C_1 can be increased. The solution sequence S_b with a feasible upper bound on C* is $C_b = \min (C_3, C_5, C_6, C_8)$, where the applicable C_v are the result of the S_v computed in the particular problem. Only two C_v will be compared in any single problem.

IV. APPLICATIONS

A. GENERAL

The algorithm developed herein is applicable in job shop scheduling to reduce the average job completion times. This will reduce the average waiting time a customer waits for his product or if the job shop processes jobs for a single customer (perhaps its own organization) then such scheduling provides for an efficient use of resources. Firms providing a repair or maintenance service for customers external to the firm can utilize the algorithm to schedule a batch of jobs with known processing times so as to reduce the average unit repair time with only a minimum number of late jobs. For organizations who own their maintenance or service facilities the utilization of the algorithm will reduce the average unit down time or average unit processing time with only a minimum number of units returned or processed beyond their desired deadlines (due-dates). The costs savings resulting from the more efficient operations should be seen in the increased production of the firms other departments.

B. COMMUNICATIONS MESSAGE HANDLING

Organizations, such as the military services, who have their own communications facilities can apply the algorithm to reduce the average delivery time of a message to addressees. This reduction which tends to expedite shorter messages is consistent with the frequently observed correlation between a message's length and its degree of urgency.

At the same time the number of late messages (with respect to desired delivery times) is kept to a minimum. Processing times are usually known in advance, especially for 100 word per minute circuits which use landline facilities operating at near 100% efficiency.

BIBLIOGRAPHY

1. Conway, R. W., Maxwell, W. L., and Miller, L. W., Theory of Scheduling (Addison-Wesley, Reading Mass., 1967).

2. Crabill, T. B., and Maxwell, W. L., "Single Machine Sequencing with Random Processing Times and Random Due-Dates," Naval Research Logistics Quarterly 16, (1969).

3. Jackson, J. R., Scheduling a Production Line to Minimize Maximum Tardiness, Management Science Research Project, UCLA (Jan. 1955).

4. Lawler, E. L., and Moore, J. M., "A Functional Equation and its Application to Resource Allocation and Sequencing Problems," Management Science, Vol. 16, No. 1 (Sept. 1969).

5. Maxwell, W. L., "On Sequencing n Jobs on One Machine to Minimize the Number of Late Jobs," Management Science, Vol. 16, No. 5, (Jan. 1970).

6. Moore, J. M., "An n Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs," Management Science, Vol. 14, No. 1, (Sept. 1968).

7. Smith, W. E., "Various Optimizers for Single-Stage Production," Naval Research Logistics Quarterly 3, (1956).

8. Sturm, L. B. J. M., "A Simple Optimality Proof of Moore's Sequencing Algorithm," Management Science, Vol. 17, No. 1, (Sept. 1970).

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Chief of Naval Personnel Pers-11b Department of the Navy Washington, D. C. 20370	1
4.	Naval Postgraduate School Department of Operations Research & Administrative Sciences Monterey, California 93940	1
5.	Assoc Professor G. T. Howard, Code 55 Hk Department of Operations Research & Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6.	CDR Richard L. Gates, USN 2924 Congress Road Pebble Beach, California 93953	1

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

An n Job, Single Machine Sequencing Algorithm for Decreasing the Sum of the Completion Times Subject to a Minimum Number of Late Jobs

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; September 1971

5. AUTHOR(S) (First name, middle initial, last name)

Richard Lee Gates

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1971	35	8

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

An n job, single machine sequencing algorithm is developed which decreases the sum of the completion times subject to a minimum number of late jobs. A primal approach is employed in which successively better solutions are obtained while maintaining feasibility. Optimality, while not claimed, may be achieved in some problems. Possible industrial and military applications are discussed.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Scheduling						
Sequencing						
Algorithm						
Optimum Job Schedules						

Thesis 130322
G2533 Gates
c.1 An n job, single machine sequencing algorithm for decreasing the sum of the completion times subject to a minimum number of late jobs.